AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph beginning on page 5, line 16, with the following rewritten paragraph:

-- Figure 1B is a block diagram of a technique used in one embodiment for capturing a security breach. In this example, technique 150 is shown to include attacker 154, Internet 158, network address translation (NAT) 162, egress traffic control 164, switch 168, database 172, honey pot server 175, honey pot server 176, honey pot management server 184, and honey pot support services 188. Honey pots 177-179 and honey pots 180-182 are virtual honey pots running on honey pot server 175 and honey pot server 176, respectively. As used herein, a virtual honey pot includes a virtual machine functioning as a honey pot. Multiple virtual honey pots can be deployed on a single server. For example, a ~~VMware~~ VMWARE server can run several virtual machines as honey pots. Each virtual honey pot can run any operating system and any number of applications. --

Please replace the paragraph beginning on page 6, line 12, with the following rewritten paragraph:

-- Each of honey pot servers 175-176 may include a management and monitoring subsystem, a data capture subsystem, and a network access control subsystem. The management and monitoring subsystem starts new honey pot instances as needed, determines when honey pots have been breached, shuts down honey pot instances, and submits captured data to automated post-intrusion analysis. The data capture subsystem captures all network data to and from a honey pot instance, and can capture data from within the honey pot instance itself if properly instrumented. The network access control subsystem drops spoofed packets from honey pot instances, cross-honey pot traffic, traffic from a honey pot to the support systems, and traffic

from a honey pot to high-risk port numbers, while also performing rate limiting of outgoing honey pot traffic. All of this can be done programmatically, for example, using ~~VMware~~ VMWARE APIs.--

Please replace the paragraph beginning on page 7, line 11, with the following rewritten paragraph:

-- Honey pot management server 184 may include a post-intrusion automated analysis subsystem that extracts files that may be associated with malicious code, such as new or modified files from a honey pot's file system. These files can be human or machine analyzed in multiple ways, depending on the goals of the user. One example of a tool that can be used for analysis is the Symantec Digital Immune System (DIS), as described in United States Patent No. 5,440,723. The subsystem can analyze captured honey pot data to detect known network based attacks, extract network flow data, and perform other analysis. Operators can configure and manage the system and access captured data using a web-based interface. For example, the web-based interface can be used to configure the following: 10 ~~Windows~~ WINDOWS NT honey pots; 5 ~~Windows~~ WINDOWS 2000 honey pots; and 2 Linux honey pots.--

Please replace the paragraph beginning on page 8, line 12, with the following rewritten paragraph:

-- In this example, a honey pot is configured (204). The honey pot can be a virtual honey pot or a physical honey pot. Examples of virtualization software that can be used to create a virtual honey pot include ~~Microsoft Virtual~~ MICROSOFT VIRTUAL PC, Bochs, and user mode Linux. The honey pot can be configured by manually installing an operating system and various applications on the honey pot. The honey pot is initially deployed (208). A physical honey pot

can be deployed by connecting it to a network. A virtual honey pot can be deployed by virtualization software, such as ~~VMware~~ VMWARE. For example, in Figure 1B, honey pot 177 is shown as deployed on honey pot server 175. The honey pot continues running until it is breached (212). A breach can be automatically detected by monitoring outgoing network traffic from the honey pot. If a breach is detected, the honey pot's state can be copied to an analysis area. The honey pot is automatically redeployed (216) so that it can immediately become available for new incidents. This can be similar or different from initially deploying the honey pot. Redeploying the honey pot can include reinitializing the state of the honey pot. The breach is analyzed (220). The analysis can be automatic. Analysis can include aggregating the data collected by or against the honey pot, such as packet dumps and IDS events. Further analysis can be performed by mounting a virtual drive and flagging any file changes. The virtual drive can be scanned for known malicious code. This information can be put in a database, which can be made available to an analyzer. The analyzer can choose to discard an incident's state, archive it, or perform further analysis. Redeploying (216) and analyzing (220) can be performed in parallel for efficiency. Optionally, analyzing (220) can be performed before redeploying the honey pot (216).--

Please replace the paragraph beginning on page 9, line 12, with the following rewritten paragraph:

-- Figure 3A is a flowchart of a technique used in one embodiment to configure a honey pot. This embodiment can be used in Figure 2 to configure a honey pot (204). In this example, software is installed on the honey pot (304). Software can be installed to make the honey pot look like a typical user's machine. For example, in Figure 1B, ~~Windows~~ WINDOWS NT, Internet Explorer, and Microsoft Exchange can be installed on honey pot 177. This configuration

could attract a Microsoft Exchange worm designed to corrupt ~~Windows~~ WINDOWS NT machines. A different operating system and different applications can be installed on honey pots 178 and 179 to attract other types of attacks. In the case of a virtual honey pot, the choice of operating systems may depend on what the virtualization software can support. An image, or master honey pot, can be created, to facilitate configuring a plurality of honey pots to run the same or similar software. For example, an image can be created that runs Linux and a particular set of applications. Multiple honey pots can then be copied from that image. Data is entered (308). For example, file system information can be entered. The number of honey pots to run for a particular image can be specified. For example, 10 honey pots can run on one particular image. Any number of honey pots can be run, depending on available resources. Information about how to handle the honey pot can be entered. The honey pot is deployed (312).--

Please replace the paragraph beginning on page 10, line 7, with the following rewritten paragraph:

-- Figure 3B is a flowchart of a technique used in one embodiment to initially deploy a honey pot. This embodiment can be used in Figure 2 to initially deploy a honey pot (208). In this example, an image is copied (318). For example, an image created during configuration of the honey pot can be copied, as described above. The image is registered with a honey pot (322). For example, the image can be registered with a virtual machine, such as a ~~VMware~~ VMWARE virtual machine. The honey pot is started (324). For example, a ~~VMware~~ VMWARE virtual machine can be instructed to start the honey pot. The configured operating system boots up and the configured applications start running. An internal IP address is assigned to the honey pot (328). For example, in Figure 1A, a DHCP service provided by honey pot support service 188 can assign the IP address.--